JOURNAL OF COMPUTER AND SYSTEM SCIENCES **53**, 575–582 (1996)
ARTICLE NO. 0088

Jerzy W. Grzymala-Busse and Soe Than

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas 66045

Received March 10, 1994; revised November 20, 1995

Data sets discussed in this paper are presented as tables with rows corresponding to examples (entities, objects) and columns to attributes. A partition triple is defined for such a table as a triple of partitions on the set of examples, the set of attributes, and the set of attribute values, respectively, preserving the structure of a table. The idea of a partition triple is an extension of the idea of a partition pair, introduced by J. Hartmanis and J. Stearns in automata theory. Results characterizing partition triples and algorithms for computing partition triples are presented. The theory is illustrated by an example of an application in machine learning from examples. © 1996 Academic Press, Inc.

1. INTRODUCTION

Reduction of data is a useful way to economize on space in current computer technology. Problems with reducing the size of data sets occur in many areas. One of them is empirical machine learning, where large input data sets make learning difficult. This paper presents a methodology for reduction of data sets, where the reduced data set preserves the structure of the original data set. Two forms of data sets are discussed in this paper. First, it is assumed that a data set is given in the form of a table, called an information system (or instance space). Rows of the table are labeled with names of examples (entities, objects), columns with names of attributes. Every example is characterized by a tuple of values of all attributes. For example, such an information system may contain data about patients in a hospital. Attributes are tests, such as surface temperature, diastolic blood pressure, systolic blood pressure. A row of the table represents a patient, characterized by a tuple of values of all tests.

The second form of a data set discussed in this paper is a decision table. A decision table is defined the same way as an information system, except that the table contains an additional column called a decision. Every example is additionally characterized by a value of the decision. A decision value is usually determined by an expert. For example, a decision table may contain hospital patient data, where a patient is characterized as being healthy or sick with some disease by an expert—a physician. This form of input data is common, e.g., for machine learning from examples.

The following problem is addressed in this paper: How to reduce the original data set to a smaller data set (containing fewer examples, attributes, and attribute values). At the same time, the smaller data set should preserve the structure of the original data set.

The proposed method of reduction of data sets is based on the idea of a partition triple, i.e., a triple of three partitions: on the set of examples, attributes, and attribute values. Every such partition clusters elements (examples, attributes, and attribute values) into blocks of elements. Additionally, examples, attributes, and attribute values are reduced into corresponding blocks in such a way that in the reduced table, where examples, attributes and attribute values are replaced by corresponding blocks, the block containing v is a value of the attribute block containing a for the example block containing x if and only if v is a value of attribute a for example x in the original table. This way the blocks of examples and attributes are transformed into blocks of attribute values in the same way that their members are transformed in the original table.

Also, a triple algebra theory, which is a basic algebraic structure for partition triples, is developed. An algorithm for computing partition triples is presented as well.

The idea of a partition triple is an extension of the idea of a partition pair, introduced by Hartmanis and Stearns in automata theory [5]. The special case of a partition triple was studied in [1]. Some preliminary results on partition triples were presented in [3, 4].

The theory of partition triples has many potential applications. One of the most obvious is relational data bases. Another application, machine learning from examples, is briefly illustrated in this paper. In this domain the input data sets are presented as decision tables. Large input data sets, representing examples, make learning difficult. This paper includes an example showing how partition triples may be used for inducing simpler rules from examples. The induced rule set represents the same knowledge as the rule set induced from the original data set.

0022-0000/96 $18.00

2. PRELIMINARY DEFINITION

Originally we will assume that the data are collected in the table, called an *information system* and defined as follows [8, 9]. The information system S is a fourtuple (E, A, V, ρ), where

E is a finite nonempty set of *examples*,

A is a finite nonempty set of *attributes*,

V is a finite nonempty set of *attribute values*, and

ρ is a function, $\rho: E \times A \to V$.

An example of such an information system is presented in Table I.

Let X be a nonempty finite set. A *partition* π on X is a family of disjoint subsets of X whose set union is X. Elements of partition π will be called *blocks* of π. If elements x and y are both members of the same block of π, it will be denoted by $x \equiv y(\pi)$. There are two trivial partitions 0_X and 1_X, where 0_X is the partition on X in which all blocks are one-element subsets of X and 1_X is the partition on X which contains only one block.

If π and τ are partitions on X, then the product of π and τ, denoted by $\pi \cdot \tau$, is a partition on X such that $x \equiv y(\pi \cdot \tau)$ if and only if $x \equiv y(\pi)$ and $x \equiv y(\tau)$. The sum of π and τ, denoted by $\pi + \tau$, is a partition on X such that $x \equiv y(\pi + \tau)$ if and only if there exists a sequence $x = x_1, x_2, ..., x_n = y$ of elements of X such that $x_i \equiv x_{i+1}(\pi)$ or $x_i \equiv x_{i+1}(\tau)$ for $i = 1, 2, ..., n-1$.

A partition π is said to be smaller than or equal to another partition τ, denoted by $\pi \leqslant \tau$, if and only if for every block B of π there exists a block B' of τ such that $B \subseteq B'$. Obviously, the product of π and τ may be defined as the greatest lower bound (g.l.b.) of π and τ, and the sum of π and τ may be defined as the least upper bound (l.u.b.) of π and τ; see, e.g. [5].

3. PARTITION TRIPLES AND MMM TRIPLES

The idea presented here of reduction data, in its special case, was originally developed in automata theory, under the same of partition pairs; see, e.g., [5]. Later on this idea, extended to input data of machine learning systems, was presented in [1]. In this paper the idea of a partition triple from [1] is generalized. In [1], in any triple of partitions, the partition on the set of all attributes was constant. Some preliminary results were discussed in [3, 4].

TABLE I

	a_1	a_2	a_3	a_4
x_1	0	0	2	2
x_2	0	1	2	2
x_3	1	1	2	3
x_4	1	1	3	3
x_5	4	3	4	4

DEFINITION. For a (E, A, V, ρ), let π be a partition on E, let τ be a partition on A, and let λ be a partition on V. A *partition triple* on an information system $S = (E, A, V, \rho)$ is an ordered triple of partitions (π, τ, λ) such that for all $x, y \in E$ and $a, b \in A$

$$x \equiv y(\pi), \quad a \equiv b(\tau) \text{ imply that } \rho(x, a) \equiv \rho(y, b)(\lambda).$$

The set of all partition triples on S will be denoted by $L(S)$.

DEFINITION. Let (π, τ, λ) be a partition triple on an information system $S = (E, A, V, \rho)$. The (π, τ, λ)-*image* of S is the information system $(\pi, \tau, \lambda, \rho')$ such that for all $B_\pi \in \pi$, $B_\tau \in \tau$, and $B_\lambda \in \lambda$

$$\rho'(B_\pi, B_\tau) = B_\lambda \qquad \text{if} \quad \rho(x, a) = v,$$

where x, a, and v are arbitrary members of B_π, B_τ, and B_λ, respectively.

An example of a partition triple of the information system from Table I is

$$(\{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}\}, \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\},$$
$$\{\{0, 1\}, \{2, 3\}, \{4\}\}).$$

The (π, τ, λ)-image of S from Table I is presented in Table II.

LEMMA 3.1. *Let (π, τ, λ) and (π', τ', λ') be partition triples on $S = (E, A, V, \rho)$, then*

(i) $(\pi \cdot \pi', \tau \cdot \tau', \lambda \cdot \lambda')$

(ii) $(\pi + \pi', \tau + \tau', \lambda + \lambda')$

are also partition triples on $S = (E, A, V, \rho)$.

Proof. Let $x, y \in E$ and $a, b \in A$.

(i) $x \equiv y(\pi \cdot \pi')$ and $a \equiv b(\tau \cdot \tau')$ implies $x \equiv y(\pi)$, $x \equiv y(\pi')$, $a \equiv b(\tau)$, and $a \equiv b(\tau')$. Since (π, τ, λ) and (π', τ', λ') are partition triples on S, $\rho(x, a) \equiv \rho(y, b)(\lambda)$, $\rho(x, a) \equiv \rho(y, b)(\lambda')$, i.e., $\rho(x, a) \equiv \rho(y, b)(\lambda \cdot \lambda')$. Therefore $(\pi \cdot \pi', \tau \cdot \tau', \lambda \cdot \lambda')$ is a partition triple on S.

TABLE II

	$\{a_1\}$	$\{a_2\}$	$\{a_3, a_4\}$
$\{x_1, x_2\}$	$\{0, 1\}$	$\{0, 1\}$	$\{2, 3\}$
$\{x_3, x_4\}$	$\{0, 1\}$	$\{0, 1\}$	$\{2, 3\}$
$\{x_5\}$	$\{4\}$	$\{2, 3\}$	$\{4\}$

(ii) Without loss of generality let us assume that $x \equiv y(\pi + \pi')$ implies that there exists a sequence $x = x_0, x_1, x_2, ..., x_n = y$ such that $x_i \equiv x_{i+1}(\pi)$ for even i and $x_i \equiv x_{i+1}(\pi')$ for odd i, where $i \in \{0, 1, ..., n-1\}$ and $a \equiv b(\tau + \tau')$ implies that there exists a sequence $a = a_0, a_1, a_2, ..., a_m = b$ such that $a_j \equiv a_{j+1}(\tau)$ for even j and $a_j \equiv a_{j+1}(\tau')$ for odd j, where $j \in \{0, 1, ..., m-1\}$. As (π, τ, λ) and (π', τ', λ') are partition triples on S, $\rho(x_i, a_0) \equiv \rho(x_{i+1}, a_0)(\lambda)$, for even i, and $\rho(x_i, a_0) \equiv \rho(x_{i+1}, a_0)(\lambda')$, for odd i. Therefore $\rho(x_0, a_0) \equiv \rho(x_n, a_0)(\lambda + \lambda')$, i.e., $\rho(x, a) \equiv \rho(y, a)(\lambda + \lambda')$. As (π, τ, λ) and (π', τ', λ') are partition triples on S, $\rho(y, a_j) \equiv \rho(y, a_{j+1})(\lambda)$, for even j, and $\rho(y, a_j) \equiv \rho(y, a_{j+1})(\lambda')$, for odd j. Therefore $\rho(y, a_0) \equiv \rho(y, a_m)(\lambda + \lambda')$, i.e., $\rho(y, a) \equiv \rho(y, b)(\lambda + \lambda')$. So, $\rho(x, a) \equiv \rho(y, b)(\lambda + \lambda')$; i.e., $(\pi + \pi', \tau + \tau', \lambda + \lambda')$ is a partition triple on S.

LEMMA 3.2. *For any partition π on E, for any partition τ on A, and for any partition λ on V, $(\pi, \tau, 1_V)$ and $(0_E, 0_A, \lambda)$ are partition triples on S.*

Proof. Let π be any partition on E, and τ be any partition on A. Let x and y be any elements of E such that $x \equiv y(\pi)$ and a and b be any elements of A such that $a \equiv b(\tau)$. Then $\rho(x, a) \equiv \rho(y, b)(1_V)$ because the partition 1_V has only one block which is the whole set V. Therefore, $(\pi, \tau, 1_V)$ is a partition triple on S.

Let λ be any partition on V. Let x and y be any elements of E such that $x \equiv y(0_E)$ and a and b be any elements of A such that $a \equiv b(0_A)$. Since each block of partitions 0_E and 0_A contains only one element, $x = y$ and $a = b$. Therefore, $\rho(x, a) = \rho(y, b)$, i.e., $\rho(x, a)$ and $\rho(y, b)$ are identical elements in V, and hence they are in the same block of any partition on V; i.e., $\rho(x, a) \equiv \rho(y, b)(\lambda)$. Therefore, $(0_E, 0_A, \lambda)$ is a partition triple on S.

LEMMA 3.3. *The set $L(S)$ of all partition triples on $S = (E, A, V, \rho)$ is a lattice.*

Proof. The proof follows directly from Lemma 3.1.

DEFINITIONS. For a given partition π on E, the minimal partition λ on V such that $(\pi, 0_A, \lambda)$ is a partition triple on $S = (E, A, V, \rho)$ will be denoted by $\mathbf{m}_{\mathbf{ev}}(\pi)$. It is obvious that

$$\mathbf{m}_{\mathbf{ev}}(\pi) = \prod \{\lambda \mid (\pi, 0_A, \lambda) \in L(S)\}.$$

Similarly, for a given partition τ on A, the minimal partition λ on V such that $(0_E, \tau, \lambda)$ is a partition triple on $S = (E, A, V, \rho)$ will be denoted $\mathbf{m}_{\mathbf{av}}(\tau)$, and

$$\mathbf{m}_{\mathbf{av}}(\tau) = \prod \{\lambda \mid (0_E, \tau, \lambda) \in L(S)\}.$$

For a given partition λ on V we may ask what are maximal partitions π on E and τ on A such that $(\pi, 0_A, \lambda)$ and $(0_E, \tau, \lambda)$ are partition triples on $S = (E, A, V, \rho)$. Such partitions will be denoted $\mathbf{M}_{\mathbf{ev}}(\lambda)$ and $\mathbf{M}_{\mathbf{av}}(\lambda)$, where

$$\mathbf{M}_{\mathbf{ev}}(\lambda) = \sum \{\pi \mid (\pi, 0_A, \lambda) \in L(S)\},$$

$$\mathbf{M}_{\mathbf{av}}(\lambda) = \sum \{\tau \mid (0_E, \tau, \lambda) \in L(S)\}.$$

In the preceding definitions, **m** stands for minimum and **M** for maximum. Partitions $\mathbf{m}_{\mathbf{ev}}(\pi)$ and $\mathbf{m}_{\mathbf{av}}(\tau)$ represent the largest amount of information about blocks of attribute values which can be drawn from the information about blocks of π and τ, respectively. Partitions $\mathbf{M}_{\mathbf{ev}}(\lambda)$ and $\mathbf{M}_{\mathbf{av}}(\lambda)$ represent the least amount of information about blocks of examples and attributes which must be supplied to identify blocks of λ.

A partition triple (π, τ, λ) on $S = (E, A, V, \rho)$ will be called a *MMm triple* if and only if

$$\pi = \mathbf{M}_{\mathbf{ev}}(\lambda), \qquad \tau = \mathbf{M}_{\mathbf{av}}(\lambda), \qquad \lambda = \mathbf{m}_{\mathbf{ev}}(\pi) + \mathbf{m}_{\mathbf{av}}(\tau).$$

The set of all MMm triples of $S = (E, A, V, \rho)$ will be denoted $K(S)$. An example of a MMm triple of the information system from Table I is

$$(\{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}, \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\},$$
$$\{\{0, 1\}, \{2, 3\}, \{4\}\}).$$

4. TRIPLE ALGEBRA

We can study properties of partition triples and MMm triples by analyzing the underlying algebraic structure, called a triple algebra. The results of the abstract structure can be applied not only to the partition triples but also to other, not yet discovered, interpretations. Let L_1, L_2, and L_3 be finite lattices. Then a subset Δ of $L_1 \times L_2 \times L_3$ is a *triple algebra* on $L_1 \times L_2 \times L_3$ if and only if the following postulates hold:

P1. (x_1, y_1, z_1) and (x_2, y_2, z_2) are in Δ implies that $(x_1 \cdot x_2, y_1 \cdot y_2, z_1 \cdot z_2)$ and $(x_1 + x_2, y_1 + y_2, z_1 + z_2)$ are in Δ,

P2. For any x in L_1, y in L_2, and z in L_3, $(x, y, 1_{L_3})$ and $(0_{L_1}, 0_{L_2}, z)$ are in Δ.

For (x, y, z) and (x', y', z') in $L_1 \times L_2 \times L_3$, we define $(x, y, z) \leqslant (x', y', z')$ if and only if $x \leqslant x'$, $y \leqslant y'$, and $z \leqslant z'$.

LEMMA 4.1. *If Δ is a triple algebra on $L_1 \times L_2 \times L_3$ and (x, y, z) is in Δ, then $x' \leqslant x$, $y' \leqslant y$, and $z' \geqslant z$ implies that (x', y', z') is in Δ.*

Proof. Suppose that (x, y, z) is in Δ and $x' \leqslant x$, $y' \leqslant y$, and $z' \geqslant z$. By the property P2, $(x', y', 1_{L_3})$ is in Δ. Hence

$(x \cdot x', y \cdot y', z \cdot 1_{L_3})$ is in Δ, by the property P1. Since $x' \leqslant x$ and $y' \leqslant y$, (x', y', z) is in Δ. By the property P2, $(0_{L_1}, 0_{L_2}, z')$ is in Δ and, hence, $(x' + 0_{L_1}, y' + 0_{L_2}, z + z')$ is in Δ. Since $z' \geqslant z$, (x', y', z') is in Δ.

DEFINITIONS. Let Δ be a triple algebra on $L_1 \times L_2 \times L_3$. For any x in L_1, y in L_2, and z in L_3, we define

$$\mathbf{m_{13}}(x) = \prod \{z \mid (x, 0_{L_2}, z) \in \Delta\},$$

$$\mathbf{m_{23}}(y) = \prod \{z \mid (0_{L_1}, y, z) \in \Delta\},$$

$$\mathbf{M_{13}}(z) = \sum \{x \mid (x, 0_{L_2}, z) \in \Delta\},$$

$$\mathbf{M_{23}}(z) = \sum \{y \mid (0_{L_1}, y, z) \in \Delta\},$$

$$\mathbf{m_{123}}(x, y) = \mathbf{m_{13}}(x) + \mathbf{m_{23}}(y),$$

$$\mathbf{M_{123}}(z) = (\mathbf{M_{13}}(z), \mathbf{M_{23}}(z)).$$

LEMMA 4.2. *For any x in L_1 and y in L_2, $\mathbf{m_{123}}(x, y) = \prod \{z \mid (x, y, z) \in \Delta\}$.*

Proof. Let us define the sets

$$R(x, y) = \{z \mid (x, y, z) \in \Delta\},$$

$$R_1(x) = \{z \mid (x, 0_{L_2}, z) \in \Delta\},$$

$$R_2(y) = \{z \mid (0_{L_1}, y, z) \in \Delta\}.$$

Then $z \in R(x, y) \Rightarrow (x, y, z) \in \Delta$

$$\Rightarrow (x, 0_{L_2}, z) \in \Delta, (0_{L_1}, y, z) \in \Delta,$$

$$\text{because } 0_{L_2} \leqslant y, 0_{L_1} \leqslant x,$$

$$\Rightarrow z \in R_1(x), z \in R_2(y).$$

Therefore, $R(x, y) \subseteq R_1(x)$ and $R(x, y) \subseteq R_2(y)$. Consequently,

$$\prod \{z \mid z \in R(x, y)\} \geqslant \prod \{z \mid z \in R_1(x)\},$$

$$\prod \{z \mid z \in R(x, y)\} \geqslant \prod \{z \mid z \in R_2(y)\};$$

i.e., $\prod \{z \mid z \in R(x, y)\} \geqslant \prod \{z \mid z \in R_1(x)\} + \prod \{z \mid z \in R_2(y)\}$, or, $\prod \{z \mid \in R(x, y)\} \geqslant \mathbf{m_{123}}(x, y)$. Let $z_1 = \prod \{z \mid z \in R_1(x)\}$ and $z_2 = \prod \{z \mid z \in R_2(y)\}$. For any $z \in R_1(x)$, $(x, 0_{L_2}, z) \in \Delta$. By the property P1, $\prod \{(x, 0_{L_2}, z) \mid z \in R_1(x)\} \in \Delta$; i.e., $(x, 0_{L_2}, z_1) \in \Delta$. Similarly, $(0_{L_1}, y, z_2) \in \Delta$. By the property P1, $(x + 0_{L_1}, 0_{L_2} + y, z_1 + z_2) \in \Delta$; i.e., $(x, y, z_1 + z_2) \in \Delta$, or $z_1 + z_2 \in R(x, y)$. Therefore $z_1 + z_2 \geqslant \prod \{z \mid z \in R(x, y)\}$; i.e., $\mathbf{m_{123}}(x, y) \geqslant \prod \{z \mid z \in R(x, y)\}$. Therefore, $\mathbf{m_{123}}(x, y) = \prod \{z \mid (x, y, z) \in \Delta\}$.

LEMMA 4.3. *For any z in L_3, $\mathbf{M_{123}}(z) = \sum \{(x, y) \mid (x, y, z) \in \Delta\}$.*

Proof. Similar to the proof of Lemma 4.2.

DEFINITION. For any two elements (x, y, z) and (x', y', z') in $L_1 \times L_2 \times L_3$, we define $(x, y, z) \leqslant (x', y', z')$ if and only if $x \leqslant x'$ in L_1, $y \leqslant y'$ in L_2, and $z \leqslant z'$ in L_3.

LEMMA 4.4. *Any triple algebra Δ on $L_1 \times L_2 \times L_3$ is a lattice under the above ordering with zero element* $(0, 0, 0)$, *unit element* $(1, 1, 1)$, *and component-wise* g.l.b. *and* l.u.b. *operations.*

DEFINITION. An element (x, y, z) in a triple algebra Δ is called a *MMm triple* if and only if $x = \mathbf{M_{13}}(z)$, $y = \mathbf{M_{23}}(z)$, and $z = \mathbf{m_{13}}(x) + \mathbf{m_{23}}(y)$. The set of all MMm triples of Δ will be denoted Q_Δ.

THEOREM 4.1. *Let Δ be a triple algebra on $L_1 \times L_2 \times L_3$. For any x in L_1, y in L_2, and z in L_3,*

(1) $(\mathbf{M_{13}}(z), 0_{L_2}, z)$, $(0_{L_1}, \mathbf{M_{23}}(z), z)$, $(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z), z)$, $(x, 0_{L_2}, \mathbf{m_{13}}(x))$, $(0_{L_1}, y, \mathbf{m_{23}}(y))$, *and* $(x, y, \mathbf{m_{123}}(x, y))$ *are in* Δ.

(2) $x_1 \leqslant x_2$ *implies* $\mathbf{m_{13}}(x_1) \leqslant \mathbf{m_{13}}(x_2)$, $y_1 \leqslant y_2$ *implies* $\mathbf{m_{23}}(y_1) \leqslant \mathbf{m_{23}}(y_2)$, *and* $x_1 \leqslant x_2$ *and* $y_1 \leqslant y_2$ *imply* $\mathbf{m_{123}}(x_1, y_1) \leqslant \mathbf{m_{123}}(x_2, y_2)$.

(3) $\mathbf{m_{13}}(x_1 + x_2) = \mathbf{m_{13}}(x_1) + \mathbf{m_{13}}(x_2)$, $\mathbf{m_{23}}(y_1 + y_2) = \mathbf{m_{23}}(y_1) + \mathbf{m_{23}}(y_2)$, *and* $\mathbf{m_{123}}(x_1 + x_2, y_1 + y_2) = \mathbf{m_{123}}(x_1, y_1) + \mathbf{m_{123}}(x_2, y_2)$.

(4) $\mathbf{m_{13}}(x_1 \cdot x_2) \leqslant \mathbf{m_{13}}(x_1) \cdot \mathbf{m_{13}}(x_2)$, $\mathbf{m_{23}}(y_1 \cdot y_2) \leqslant \mathbf{m_{23}}(y_1) \cdot \mathbf{m_{23}}(y_2)$, *and* $\mathbf{m_{123}}(x_1 \cdot x_2, y_1 \cdot y_2) \leqslant \mathbf{m_{123}}(x_1, y_1) \cdot \mathbf{m_{123}}(x_2, y_2)$.

(5) $z \geqslant \mathbf{m_{13}}(x)$ *if and only if* $(x, 0_{L_2}, z) \in \Delta$, $z \geqslant \mathbf{m_{23}}(y)$ *if and only if* $(0_{L_1}, y, z) \in \Delta$, *and* $z \geqslant \mathbf{m_{123}}(x, y)$ *if and only if* $(x, y, z) \in \Delta$.

(6) $z_1 \leqslant z_2$ *implies that* $\mathbf{M_{13}}(z_1) \leqslant \mathbf{M_{13}}(z_2)$ *and* $\mathbf{M_{23}}(z_1) \leqslant \mathbf{M_{23}}(z_2)$.

(7) $\mathbf{M_{13}}(z_1 + z_2) \geqslant \mathbf{M_{13}}(z_1) + \mathbf{M_{13}}(z_2)$ *and* $\mathbf{M_{23}}(z_1 + z_2) \geqslant \mathbf{M_{23}}(z_1) + \mathbf{M_{23}}(z_2)$.

(8) $\mathbf{M_{13}}(z_1 \cdot z_2) = \mathbf{M_{13}}(z_1) \cdot \mathbf{M_{13}}(z_2)$ *and* $\mathbf{M_{23}}(z_1 \cdot z_2) = \mathbf{M_{23}}(z_1) \cdot \mathbf{M_{23}}(z_2)$.

(9) $x \leqslant \mathbf{M_{13}}(z)$ *if and only if* $(x, 0_{L_2}, z) \in \Delta$, $y \leqslant \mathbf{M_{23}}(z)$ *if and only if* $(0_{L_1}, y, z) \in \Delta$, *and* $x \leqslant \mathbf{M_{13}}(z)$ *and* $y \leqslant \mathbf{M_{23}}(z)$ *if and only if* $(x, y, z) \in \Delta$.

(10) $\mathbf{M_{13}}(\mathbf{m_{13}}(x)) \geqslant x$, $\mathbf{M_{23}}(\mathbf{m_{23}}(y)) \geqslant y$, $\mathbf{M_{13}}(\mathbf{m_{123}}(x, y)) \geqslant x$, *and* $\mathbf{M_{23}}(\mathbf{m_{123}}(x, y)) \geqslant y$.

(11) $\mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z)) \leqslant z$, $\mathbf{m_{13}}(\mathbf{M_{13}}(z)) \leqslant z$, *and* $\mathbf{m_{23}}(\mathbf{M_{23}}(z)) \leqslant z$.

(12) $\mathbf{M_{13}}(\mathbf{m_{13}}(\mathbf{M_{13}}(z))) = \mathbf{M_{13}}(z)$, $\mathbf{M_{23}}(\mathbf{m_{23}}(\mathbf{M_{23}}(z))) = \mathbf{M_{23}}(z)$, $\mathbf{M_{13}}(\mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z))) = \mathbf{M_{13}}(z)$, *and* $\mathbf{M_{23}}(\mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z))) = \mathbf{M_{23}}(z)$.

(13) $\mathbf{m_{13}}(\mathbf{M_{13}}(\mathbf{m_{13}}(x))) = \mathbf{m_{13}}(x)$, $\mathbf{m_{23}}(\mathbf{M_{23}}(\mathbf{m_{23}}(y))) = \mathbf{m_{23}}(y)$, *and* $\mathbf{m_{123}}(\mathbf{M_{13}}(\mathbf{m_{123}}(x, y))$, $\mathbf{M_{23}}(\mathbf{m_{123}}(x, y))) = \mathbf{m_{123}}(x, y)$.

(14) $(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z), \mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z)))$ *and* $(\mathbf{M_{13}}(\mathbf{m_{123}}(x, y)), \mathbf{M_{23}}(\mathbf{m_{123}}(x, y)), \mathbf{m_{123}}(x, y))$ *are in* Q_Δ.

(15) *If* (x_1, y_1, z_1) *and* (x_2, y_2, z_2) *are in* Q_Δ, *then* $x_1 \leqslant x_2$ *and* $y_1 \leqslant y_2$ *if and only if* $z_1 \leqslant z_2$.

(16) *The set* Q_Δ *under the ordering* Δ *is a lattice in which*

$$\begin{aligned}&\text{g.l.b.}\{(x_1, y_1, z_1), (x_2, y_2, z_2)\}\\ &\quad = (x_1 \cdot x_2, y_1 \cdot y_2, \mathbf{m_{123}}(x_1 \cdot x_2, y_1 \cdot y_2)),\\ &\text{l.u.b.}\{(x_1, y_1, z_1), (x_2, y_2, z_2)\}\\ &\quad = (\mathbf{M_{13}}(z_1 + z_2), \mathbf{M_{23}}(z_1 + z_2), z_1 + z_2).\end{aligned}$$

Proof. (1) By the definitions of $\mathbf{M_{13}}$ and $\mathbf{M_{23}}$, $(\mathbf{M_{13}}(z), 0_{L_2}, z)$ and $(0_{L_1}, \mathbf{M_{23}}(z), z)$ are in Δ. By the property P1, $(\mathbf{M_{13}}(z) + 0_{L_1}, 0_{L_2} + \mathbf{M_{23}}(z), z + z)$ is in Δ; i.e., $(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z), z)$ is in Δ. Similarly, $(x, 0_{L_2}, \mathbf{m_{13}}(x))$ and $(0_{L_1}, y, \mathbf{m_{23}}(y))$ are in Δ and so is their sum; i.e., $(x + 0_{L_1}, 0_{L_2} + y, \mathbf{m_{13}}(x) + \mathbf{m_{23}}(y))$ is in Δ; i.e., $(x, y, \mathbf{m_{123}}(x, y))$ is in Δ.

(2)–(16) are either similar to the proof of (1) or straightforward.

The following result gives characterization of Δ in terms of Q_Δ.

THEOREM 4.2. *Let* Δ *be a triple algebra on* $L_1 \times L_2 \times L_3$. *Let* x *in* L_1, y *in* L_2, *and* z *in* L_3. *Then* (x, y, z) *is in* Δ *if and only if there exists* (x', y', z') *in* Q_Δ *such that* $x \leqslant x'$, $y \leqslant y'$, *and* $z \geqslant z'$.

Proof. Suppose that (x, y, z) is in Δ. Let $x' = \mathbf{M_{13}}(z)$, $y' = \mathbf{M_{23}}(z)$, and $z' = \mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z))$. By Theorem 4.1.9, $x \leqslant \mathbf{M_{13}}(z)$ and $y \leqslant \mathbf{M_{23}}(z)$; i.e., $x \leqslant x'$ and $y \leqslant y'$. By Theorem 4.1(11), $\mathbf{m_{123}}(\mathbf{M_{13}}(z), \mathbf{M_{23}}(z)) \leqslant z$; i.e., $z' \leqslant z$. By Theorem 4.1(14), (x', y', z') is in Q_Δ. Therefore, there exists (x', y', z') in Q_Δ such that $x \leqslant x'$, $y \leqslant y'$, and $z \geqslant z'$.

Now suppose that there exists (x', y', z') in Q_Δ such that $x \leqslant x'$, $y \leqslant y'$, and $z \geqslant z'$. Since $Q_\Delta \subseteq \Delta$, (x', y', z') is in Δ and, hence, (x, y, z) is in Δ, by Lemma 4.1.

LEMMA 4.5. *If* (x_1, y_1, z_1) *and* (x_2, y_2, z_2) *are in* Q_Δ, *then the following three statements are equivalent*:

(1) $(x_1, y_1, z_1) \geqslant (x_2, y_2, z_2)$,

(2) $x_1 \geqslant x_2$ *and* $y_1 \geqslant y_2$,

(3) $z_1 \geqslant z_2$.

Proof. By Theorem 4.1(15), statement (2) and statement (3) are equivalent. By the definition of ordering relation in $L_1 \times L_2 \times L_3$, statement (1) is equivalent to the combination of statements (2) and (3). Therefore, the three statements are equivalent.

LEMMA 4.6. *If* (x, y, z) *is in* Δ, *then* $(\mathbf{M_{13}}(z), y, z)$ *and* $(x, \mathbf{M_{23}}(z), z)$ *are also in* Δ.

Proof. Assume that (x, y, z) is in Δ. Then $(0_{L_1}, y, z)$ is in Δ, by Lemma 4.1. By Theorem 4.1(1), $(\mathbf{M_{13}}(z), 0_{L_2}, z)$ is in Δ. Therefore, $(0_{L_1} + \mathbf{M_{13}}(z), y + 0_{L_2}, z + z)$ is in Δ; i.e., $(\mathbf{M_{13}}(z), y, z)$ is in Δ. Similarly, $(x, \mathbf{M_{23}}(z), z)$ is in Δ.

LEMMA 4.7. $\mathbf{M_{13}}(\mathbf{m_{23}}(\mathbf{M_{23}}(z))) \leqslant \mathbf{M_{13}}(z)$ *and* $\mathbf{M_{23}}(\mathbf{m_{13}}(\mathbf{M_{13}}(z))) \leqslant \mathbf{M_{23}}(z)$.

Proof. By Theorem 4.1(11), $\mathbf{m_{23}}(\mathbf{M_{23}}(z)) \leqslant z$. By Theorem 4.1(6), $\mathbf{M_{13}}(\mathbf{m_{23}}(\mathbf{M_{23}}(z))) \leqslant \mathbf{M_{13}}(z)$. Similarly, $\mathbf{M_{23}}(\mathbf{m_{13}}(\mathbf{M_{13}}(z))) \leqslant \mathbf{M_{23}}(z)$.

LEMMA 4.8. $\mathbf{m_{13}}(\mathbf{M_{13}}(\mathbf{m_{23}}(y))) \leqslant \mathbf{m_{23}}(y)$ *and* $\mathbf{m_{23}}(\mathbf{M_{23}}(\mathbf{m_{13}}(x))) \leqslant \mathbf{m_{13}}(x)$.

Proof. By Theorem 4.1(11), $\mathbf{m_{13}}(\mathbf{M_{13}}(z)) \leqslant z$, for any z in L_3. Since $\mathbf{m_{23}}(y)$ is in L_3, $\mathbf{m_{13}}(\mathbf{M_{13}}(\mathbf{m_{23}}(y))) \leqslant \mathbf{m_{23}}(y)$. Similarly, $\mathbf{m_{23}}(\mathbf{M_{23}}(\mathbf{m_{13}}(x))) \leqslant \mathbf{m_{13}}(x)$.

THEOREM 4.3. *The set* $L(S)$ *of all partition triples on* $S = (E, A, V, \rho)$ *is a triple and, hence, satisfies the above propositions.*

Proof. By Lemma 3.2 and by Lemma 3.1, the set $L(S)$ satisfies the properties P1 and P2, respectively, of the triple algebra.

Therefore, $L(S)$ is a triple algebra. The definitions $\mathbf{m_{ev}}$, $\mathbf{m_{av}}$, $\mathbf{M_{ev}}$, and $\mathbf{M_{av}}$ of the partitions are analogous to the definitions $\mathbf{m_{13}}$, $\mathbf{m_{23}}$, $\mathbf{M_{13}}$, and $\mathbf{M_{23}}$, respectively, of the triple algebra. So all the results on the triple algebra can be applied to partition triples by replacing $\mathbf{m_{13}}$, $\mathbf{m_{23}}$, $\mathbf{M_{13}}$, $\mathbf{M_{23}}$, Δ, and Q_Δ with $\mathbf{m_{ev}}$, $\mathbf{m_{av}}$, $\mathbf{M_{ev}}$, $\mathbf{M_{av}}$, $L(S)$, and $K(S)$, respectively.

Thus, by Theorem 4.2, any partition triple on S can be computed from an MMm triple on S by refining the first two partitions and coarsening the third partition of the MMm triple. Also, it is sufficient to compute set $K(S)$ and then compute $L(S)$ from $K(S)$.

5. ALGORITHM FOR COMPUTING *K*(*S*)

An algorithm to determine all MMm triples is an extension of the algorithm to determine Mm pairs for automata [5]. Let $\pi_{x,y}$ denote the partition on E such that all blocks of $\pi_{x,y}$ except one, are singletons, and the only block of $\pi_{x,y}$ that is not a singleton contains two elements: x and y. Similarly, let $\tau_{a,b}$ denote the partition on A such that all blocks of $\tau_{a,b}$, except one, are singletons, and the only block of $\tau_{a,b}$ that is not a singleton contains two elements: a and b. Our algorithm is based on the following result.

THEOREM 5.1. *If (π, τ, λ) is a MMm triple then*

$$\lambda = \sum \{\mathbf{m_{ev}}(\pi_{x,y} \mid \pi_{x,y} \leqslant \pi\} + \sum \{\mathbf{m_{av}}(\tau_{a,b}) \mid \tau_{a,b} \leqslant \tau\}.$$

Proof. First, $\pi \geqslant \pi_{x,y}$ and $\tau \geqslant \tau_{a,b}$; hence $(\pi_{x,y}, 0_A, \lambda)$ and $(0_E, \tau_{a,b}, \lambda)$ are partition triples, $\mathbf{m_{ev}}(\pi_{x,y}) \leqslant \lambda$ and $\mathbf{m_{av}}(\tau_{a,b}) \leqslant \lambda$. Then $\sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\} \leqslant \lambda$ and $\sum \{\mathbf{m_{av}}(\tau_{a,b}) \mid \tau_{a,b} \leqslant \tau\} \leqslant \lambda$. Hence,

$$\sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\} + \sum \{\mathbf{m_{av}}(\tau_{a,b} \leqslant \tau\} \leqslant \lambda.$$

On the other hand, $(\sum \{\pi_{x,y} \mid \pi_{x,y} \leqslant \pi\}, 0_A, \sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\})$ and $(0_E, \sum \{\tau_{a,b} \mid \tau_{a,b} \leqslant \tau\}, \sum \{\mathbf{m_{av}}(\tau_{a,b}) \mid \tau_{a,b} \leqslant \tau\})$ are partition triples and so is $(\sum \{\pi_{x,y} | \pi_{x,y} \leqslant \pi\}, \sum \{\tau_{a,b} | \tau_{a,b} \leqslant \tau\}, \sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\} + \sum \{\mathbf{m_{av}}(\tau_{a,b}) \mid \tau_{a,b} \leqslant \tau\}) = (\pi, \tau, \sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\} + \sum \{\mathbf{m_{av}}(\tau_{a,b} \mid \tau_{a,b} \leqslant \tau\})$. Therefore,

$$\sum \{\mathbf{m_{ev}}(\pi_{x,y}) \mid \pi_{x,y} \leqslant \pi\} + \sum \{\mathbf{m_{av}}(\tau_{a,b}) \mid \tau_{a,b} \leqslant \tau\}$$
$$\geqslant \mathbf{m_{ev}}(\pi) + \mathbf{m_{av}}(\tau) = \lambda,$$

and the result is proved.

Since $\pi_{x,y}$ and $\tau_{a,b}$ are the smallest nontrivial partitions on E and A, respectively, partitions $\mathbf{m_{ev}}(\pi_{x,y})$ and $\mathbf{m_{av}}(\tau_{a,b})$ are the smallest m-type partitions on V.

In the first stage of the algorithm sets $R_1 = \{\mathbf{m_{ev}}(\pi_{x,y}) \mid (x, y) \in E \times E\}$ and $R_2 = \{\mathbf{m_{av}}(\tau_{a,b}) \mid (a, b) \in A \times A\}$ are computed. For computation of R_1 only $|E| \cdot (|E| - 1)/2$ steps are required, because $\mathbf{m_{ev}}(\pi_{x,y}) = \mathbf{m_{ev}}(\pi_{y,x})$ and $\mathbf{m_{ev}}(\pi_{x,x}) = 0_V$, where $|X|$ denotes the cardinality of the set X. Similarly, computation of the set R_2 requires only $|A| \cdot (|A| - 1)/2$ steps.

In the second stage of the algorithm, the set

$$R^{(1)} = \{\lambda + \lambda' \mid \lambda \in R_1, \lambda' \in R_2\}$$

is computed. Obviously, $R_1 \subseteq R^{(1)}$ and $R_2 \subseteq R^{(1)}$, because $0_V \in R_1$ and $0_V \in R_2$. Then the set

$$R^{(2)} = \{\lambda + \lambda' \mid \lambda \in R^{(1)}, \lambda' \in R^{(1)}\}$$

should be computed. Also, $R^{(1)} \subseteq R^{(2)}$. Similarly, the set $R^{(k+1)}$ is computed from $R^{(k)}$ by

$$R^{(k+1)} = \{\lambda + \lambda' \mid \lambda \in R^{(k)}, \lambda' \in R^{(k)}\}.$$

The process stops when, for some k, $R^{(k)} = R^{(k+1)} = R$. Thus, the set $R^{(1)}$ is the set of generators for R. Every m-type partition $\lambda \in R$ determines two unique M-type partitions on E and A, respectively. These M-type partitions are $\mathbf{M_{ev}}(\lambda)$ and $\mathbf{M_{av}}(\lambda)$, respectively. Moreover, for any $\lambda \in R$, $\mathbf{M_{ev}}(\lambda)$ and $\mathbf{M_{av}}(\lambda)$ may be computed using the following formulas:

$$\mathbf{M_{ev}}(\lambda) = \sum \{\pi_{x,y} \mid \mathbf{m_{ev}}(\pi_{x,y}) \leqslant \lambda\},$$

$$\mathbf{M_{av}}(\lambda) = \sum \{\tau_{a,b} \mid \mathbf{m_{av}}(\tau_{a,b}) \leqslant \lambda\}.$$

Finally, the set $K(S)$ of all MMm triples is

$$\{(\mathbf{M_{ev}}(\lambda), \mathbf{M_{av}}(\lambda), \lambda) \mid \lambda \in R\}.$$

6. DECISION TABLES

In this section we will assume that the data sets are presented in the form of a decision table. The following definition of a decision table is a slightly modified version of the definition introduced by Pawlak [8, 9]. The decision table T is a sixtuple (E, A, V, d, W, ρ), where

E is a finite nonempty set of *examples*,

A is a finite nonempty set of *attributes*,

V is a finite nonempty set of *attribute values*,

d is a variable called a *decision*,

W is a finite nonempty set of decision values,

$\rho: E \times (A \cup \{d\}) \to V \cup W$, where if ρ is restricted to $E \times A$ it has values from V, and if ρ is restricted to $E \times \{d\}$ it has values from W.

For the sake of simplicity, restrictions of ρ to $E \times A$ and to $E \times \{d\}$ will also be denoted ρ.

An example of the decision table is presented in Table III.

DEFINITION. For a decision table (E, A, V, d, W, ρ), let π be a partition on E, let τ be a partition on A, and let λ be a partition on V. A *partition triple* on a decision table $T = (E, A, V, d, W, \rho)$ is an ordered triple of partitions (π, τ, λ) such that for all $x, y \in E$ and $a, b \in A$:

$$x \equiv y(\pi),\ a \equiv b(\tau) \text{ imply that } \rho(x, a) \equiv \rho(y, b)(\lambda),$$
$$\rho(x, d) = \rho(y, d).$$

The set of all partition triples on T will be denoted by $L(S)$.

TABLE III

	Attributes				Decision
	a_1	a_2	a_3	a_4	d
x_1	0	0	2	2	0
x_2	0	1	2	2	0
x_3	1	1	2	3	1
x_4	1	1	3	3	1
x_5	4	3	4	4	1

DEFINITION. Let (π, τ, λ) be a partition triple on a decision table $T=(E, A, V, d, W, \rho)$. The (π, τ, λ)-*image* is the decision table $(\pi, \tau, \lambda, d, W, \rho')$ such that for all $B_\pi \in \pi$, $B_\tau \in \tau$, and $B_\lambda \in \lambda$,

$$\rho'(B_\pi, B_\tau) = B_\lambda, \rho'(B_\pi, d) = \rho(x, d) \qquad \text{if} \quad \rho(x, a) = v,$$

where x, a, and v are arbitrary members of B_π, B_τ, and B_λ, respectively.

LEMMA 6.1. *Let* (π, τ, λ) *and* (π', τ', λ') *be partition triples on* $T=(E, A, V, d, W, \rho)$, *then*

(i) $(\pi \cdot \pi', \tau \cdot \tau', \lambda \cdot \lambda')$,

(ii) $(\pi + \pi', \tau + \tau', \lambda + \lambda')$

are also partition triples on $T=(E, A, V, d, W, \rho)$.

Proof. Straightforward extension of the proof of Lemma 3.1.

LEMMA 6.2. *For any partition* π *on* E *such that* $\pi \leqslant \{d\}^*$, *for any partition* τ *on* A *and for any partition* λ *on* V, $(\pi, \tau, 1_V)$ *and* $(0_E, 0_A, \lambda)$ *are partition triples on* S.

Proof. Straightforward extension of the proof of Lemma 3.2.

LEMMA 6.3. *The set* $L(T)$ *of all partition triples on* $T=(E, A, V, d, W, \rho)$ *is a lattice.*

Proof. Follows directly from Lemma 6.1.

Operators $\mathbf{m_{ev}}$, $\mathbf{m_{av}}$, $\mathbf{M_{ev}}$, and $\mathbf{M_{av}}$ for decision tables may be defined in the same way as for information systems. Moreover, all previous results of the triple algebra are valid for decision tables as well. In particular, the algorithm for computing the set $K(S)$ of all MMm triples for information systems may be used for decision tables with little changes.

7. APPLICATIONS—AN EXAMPLE

There are many possible applications of the theory presented. In any area where information systems or decision tables are used, the obvious benefits of simplification can be utilized. One of the evident areas of applications is relational data bases. Another area, less evident, is machine learning from examples.

Reduction of input data sets in machine learning from examples may be considered a kind of preprocessing. Other known approaches to preprocessing of input data in machine learning include selecting the most representative examples [6, 7] and a kind of refinement [10].

Let us illustrate the application of partition triple theory to machine learning from examples using the example of input data in the form of a decision table from Table IV.

Using machine learning system LERS [2], the following rules were induced:

(Quantitative, Excellent) & (Reading, Excellent)

→ (Admission, Accept),

(Grammar, Excellent) & (Quantitative, High)

→ (Admission, Accept),

(Advanced, Excellent) → (Admission, Accept),

(Grammar, Medium) → (Admission, Reject),

(Grammar, Low) → (Admission, Reject),

(Quantitative, Low) → (Admission, Reject),

(Quantitative, Medium) → (Admission, Reject).

One of the partition triples of the decision table from Table IV is

$$(\{\{x_1, x_2, x_3\}, \{x_4, x_5\}, \{x_6, x_7\}, \{x_8, x_9\}\},$$
$$\{\{\text{Quantitative, Analytical}\}, \{\text{Advanced}\},$$
$$\{\text{Grammar, Reading}\}\},$$
$$\{\{\text{Excellent, High}\}, \{\text{Medium, Low}\}\}).$$

TABLE IV

	Attributes					Decision
	Quantitative	Analytical	Advanced	Grammar	Reading	Admission
x_1	High	Excellent	High	Excellent	Excellent	Accept
x_2	Excellent	High	Excellent	High	High	Accept
x_3	Excellent	High	Excellent	High	Excellent	Accept
x_4	High	Excellent	Medium	Excellent	High	Accept
x_5	Excellent	High	Low	High	Excellent	Accept
x_6	Excellent	High	Medium	Medium	Low	Reject
x_7	High	Excellent	Low	Low	Medium	Reject
x_8	Low	Medium	Medium	Excellent	High	Reject
x_9	Medium	Low	Low	High	Excellent	Reject

TABLE V

	Attributes			Decision
	Aptitude	Advanced	Language	Admission
x_1	Above_avg	Above_avg	Above_avg	Accept
x_4	Above_avg	Below_avg	Above_avg	Accept
x_6	Above_avg	Below_avg	Below_avg	Reject
x_8	Below_avg	Below_avg	Above_avg	Reject

After assigning new names for the blocks of attributes and for the blocks of attribute values, the corresponding reduced decision table is presented in Table V.

The rules induced by LERS from the reduced decision table are:

$$(\text{Aptitude, Above\_avg}) \,\&\, (\text{Language, Above\_avg}) \rightarrow (\text{Admission, Accept}),$$

$$(\text{Aptitude, Below\_avg}) \rightarrow (\text{Admission, Reject}),$$

$$(\text{Language, Below\_avg}) \rightarrow (\text{Admission, Reject}).$$

The above set of rules represents exactly the same knowledge as the set of rules induced from the original decision table, yet this set is much simpler and more evident.

8. CONCLUSIONS

The theory of partition triples of data sets is presented in the paper mostly for information systems resembling relational databases. However, all results, with respective changes, are valid for decision tables as well. The theory may be used in an obvious way—for computing simpler data sets, while preserving the structure of the original data sets. The main idea is to compute the set K of all MMm triples of a data set. Any partition triple may be computed from a suitable member of K by refining the first two partitions and coarsening the third partition. The theory is illustrated by an example of application from the area of machine learning, showing that induced rules from the simplified data are more evident.

The disadvantage of the presented algorithms is their computational complexity. In general, the worst-case time computational complexity for the algorithms to compute all MMm partitions is exponential. Therefore, new, less complex algorithms should be developed, producing only some partition triples, perhaps even only one good partition triple.

ACKNOWLEDGMENT

The authors thank the anonymous referee for many valuable comments on the presentation.

REFERENCES

1. J. W. Grzymala-Busse, On the reduction of instance space in learning from examples, *in* "Proceedings, 5th International Symposium on Methodologies for Intelligent Systems, Knoxville, TN, October 1990," pp. 388–395, North Holland, New York, 1991.
2. J. W. Grzymala-Busse, LERS—A system for learning from examples based on rough sets, *in* "Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory" (R. Slowinski, Ed.), pp. 3–18, Kluwer Academic, Dordrecht, 1992.
3. J. W. Grzymala-Busse and S. Than, Reduction of instance space in machine learning from examples, *in* "Proceedings, 5th International Symposium on Artificial Intelligence, Cancun, Mexico, December 1992," pp. 303–309.
4. J. W. Grzymala-Busse and S. Than, Data compression in machine learning applied to natural language, *Behav. Res. Methods Instrum. Comput.* **25** (1993), 318–321.
5. J. Hartmanis and R. E. Stearns, "Algebraic Structure Theory of Sequential Machines," Prentice–Hall, Englewood Cliffs, NJ, 1966.
6. D. Kibler and D. W. Aha, Learning representative exemplars of concepts: An initial case study, *in* "Proceedings, 4th International Workshop on Machine Learning, Irvine, CA, June 1987," pp. 24–30.
7. R. S. Michalski and R. L. Chilausky, Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology, *Int. J. Man-Mach. Studies* **12** (1980), 63–87.
8. Z. Pawlak, Rough sets, *Int. J. Comput. Inform. Sci.* **11** (1982), 341–356.
9. Z. Pawlak, Rough classification, *Int. J. Man-Mach. Studies* **20** (1984), 469–483.
10. W. Van de Velde, Learning through progressive refinement, *in* "Proceedings, EWSL 88, 3rd European Working Session on Learning, Glasgow, United Kingdom, October 1988," pp. 211–216.